INTERNET KEY EXCHANGE VERSION 2 WITH IPV6 CRYPTOGRAPHICALLY GENERATED ADDRESSES

Jean-Michel Combes, Aurélien Wailly
France Telecom - Orange Labs, Issy-Les-Moulineaux, France
Maryline Laurent
Institut TELECOM - TELECOM SudParis, Evry, France

Internet Protocol security (IPsec) is a protocol suite enabling secure IP communications by authentication and/or encryption. Even if, IPsec may be manually configured on IP nodes, e.g., for a simple architecture, *Internet Key Exchange version 2* (IKEv2) is generally used to make deployment easier. Authentication of each peer is usually based on one of the following security materials: pre-shared keys, X.509 certificates and *Extensible Authentication Protocol* (EAP). However, these methods may have drawbacks. On the other hand, *Cryptographically Generated Addresses* (CGA) are IPv6 addresses where the last 64 bits are the hash computation over the concatenation of a public key and specific parameters. Such a type of addresses is the main element of the mechanism to secure the IPv6 Neighbor Discovery protocol, but where the CGA security proprieties are only used in a local scope (i.e., on the link where the CGA owner is located). An interesting solution could be the use of CGA as alternative security material for IKEv2, as submitted in a previous academic paper and an *Internet Engineering Task Force* (IETF) proposal. Our aims were to analyze advantages and drawbacks of CGA use compared to classical IKEv2 security materials, to decide design choices regarding modifications of IKEv2 to integrate CGA, and finally, to implement and test them.

At first, EAP support is not mandatory in IKEv2 specifications and so interoperability issues may appear. Pre-shared keys are generally not used with IKEv2 because provision of them is complex and generally not scalable. X.509 certificates generally require to deploy a dedicated architecture, like a *Public Key Infrastructure* (PKI), which may be heavy to manage and messages exchanges between PKI entities can introduce critical threats when one element of the trust chain is compromised. CGA doesn't need a certification chain, and so a dedicated infrastructure, unlike X.509 certificates built on a PKI, because the CGA is generated by its owner and all the security material needed to check the CGA, and the associated message, is sent directly to the message receiver. Now, self-signed certificates, which only require a public/private keys pair and no infrastructure like CGA, can be generated by its owner but it is impossible to check whether the information within the certificate are correct or not. As such, CGA are more secure as the attacker should break hash function and find a collision to get the same level of security.

Concerning the drawbacks, at first, CGA, as an IPv6 address, is hard for a human to be remembered (especially an IPv6 address which is longer than an IPv4 address and hexadecimal encoded). CGA may be used also for anonymity (i.e., no link between the layer 3 address and the layer 2 address) and so could

change frequently. Thus, it should be necessary to associate a CGA to a *Fully Qualified Domain Name* (FQDN) and to store this information in a *Domain Name Server* (DNS). But as DNS exchanges may be the subject of attacks, e.g., either during DNS dynamic updates or DNS requests, this should result to secure the DNS exchanges to keep the security level provided by CGA. Another drawback is that CGA specifications only allow the use of two cryptographic algorithms: the public-key algorithm RSA and the hash function SHA-1. RSA is not well adapted for constrained nodes like mobiles or sensors and SHA-1, based on recent cryptanalysis, should be broken in a near future. Finally, a CGA cannot be revoked like a X.509 certificate when the CGA has been compromised by a collision.

To integrate CGA in IKEv2, IKEv2 payloads and messages exchanges need to be modified. The CGA must be now the identity used during IKEv2 exchanges, CGA parameters are carried as a self-signed certificate and, finally, the authentication is based on the private key associated to the public key from the CGA parameters.

The proof of concept is based on two current implementations: the CGA library developed by DoCoMo USA Labs and the IKEv2 implementation on Linux named StrongSwan. Many modifications have been done on this last one. At first, the IPsec configuration file and the associated parser have been modified to accept CGA based rules. As said before, the CGA must be now the identity used during IKEv2 exchanges and corresponding CGA parameters, encoded in a X.509 certificate template, are provided during these exchanges. The default parser inside StrongSwan can dynamically include functions via plugins, so, to be able to parse the CGA Parameters inside this certificate and to transform them into a valid format for StrongSwan, a new plugin was implemented. Regarding the IKEv2 authentication process, now, before checking the validity of the signature, the IKEv2 daemon must verify the validity of the CGA (i.e., the IKEv2 daemon re-generates the CGA using the provided CGA parameters as specified in the CGA specifications). Finally, to check the information were correctly provided during IKEv2 exchanges, a plugin for Wireshark was implemented.

From the analysis results, CGA use with IKEv2 has major advantages in comparaison to classical IKEv2 security materials. Now, most of the drawbacks can be solved. The security of DNS exchanges can be provided by *DNS Security* (DNSSEC) and, either *Secret Key Transaction Authentication for DNS* (TSIG) or *DNS Request and Transaction Signatures* (SIG(0)). Concerning the limitation about RSA, recent proposals using Elliptic Curve based algorithms have been submitted. For SHA-1, with SHA-3 future adoption, CGA standard should be consequently updated. Now, CGA revocation is still an open issue and works for a potential solution are still needed. Based on our design choices, we have an implementation allowing to set up IPsec connections based on CGA and, as far as we know, this is the only implementation working with IKEv2

on Linux. This could help us to move forward such a solution to the IETF standardisation organism.